

Specification

Please amend paragraph [0003] to read as follows:

[0003] While object oriented programming has several advantages, it can be a difficult and time consuming process for programmers to develop complex object oriented applications. The programmer must first determine the parameters of the application desired (e.g., input variable, desired screen appearance, output variables, etc.). An appropriate format or language is then selected to create the application (e.g., gif, Java JAVA, J2EE, C/C++, etc.). Then the programmer must draft the programming code using the desired format in order to create the application or desired objects. This can be a time intensive, detailed process.

Please amend paragraph [0006] to read as follows:

[0006] The present invention provides a method, system, and computer program product for automatically developing objects using a plurality of context derived models (e.g., XML, [[.]]gif, Java JAVA, C/C++) residing within a computational grid. An object meta language (OML) is used to allow a programmer to define an application. Using OML, the programmer creates a document describing the required object. The OML document is submitted to a group of context derived models residing at various computational nodes on the grid. A web service is used to parse the OML document and select the appropriate node. The OML document is provided to the selected node, which applies object description variables using a transform language, such as eXtensible Style Language (XSL). The defined object is then returned to the

programmer, thus eliminating the need for the programmer to generate the actual code for the desired object.

Please amend paragraph [0013] to read as follows:

[0013] Fig. 2 is a flow chart illustrating an example of steps performed in accordance with the method of the present invention. At step 21, a programmer who desires to create an application describes the desired application in a highly abstract form. In an exemplary embodiment, ~~an a~~ meta language is used to allow the programmer to build a description with a high level of abstractions. In an exemplary embodiment, the object meta language (OML) is an XML dialect containing meta tags that can be parsed using XML technologies. Other languages, however, could also be used to set forth the initial definition. ~~[[.]]~~

Please amend paragraphs [0015] through [0016] to read as follows:

[0015] At step 23, the OML document is provided to a web service. The web service parses the document using XML technologies to determine a suitable code generation module. At step 25, the web service surveys all available grid nodes in order to locate an available node that comprises a style sheet or XML template in accordance with the application parameters set forth in the OML document. In an exemplary embodiment, the grid nodes contain XSL style sheets or XML templates capable of generating completely coded applications from XML definitions. Alternatively, the nodes could contain simple gif generation modules or more complex applications such as Java

JAVA application, J2EE applications, or C/C++ applications. The web service selects the appropriate module based upon the OML definitions set forth by the author.

[0016] Once a suitable style sheet or template is located, the web service provides the parsed OML definition to the selected node (step 27). The node on the grid uses a context derived model to generate the code for the desired application. For example, a particular node may contain an XSL style sheet to perform an XSL/XML transform. An XML/XSL transform is used to take XML application definitions and create a fully described application by applying a predetermined style template to the parsed XML code. For example, an author might express a desire for the application to display a title by using an XML <title> tag. The XSL module takes this information and creates a styled title definition (e.g., a particular bold faced font, a shaped word configuration, a word/motion combination). The models residing on the various nodes within the grid are able to create various output formats, depending upon the application desired. For example, after the OML definition is parsed by the node and the XML/XSL transfer is applied, the output from the XML/XSL transform can be an XML document, a Java Server Page JAVA SERVER PAGE (JSP), a Java JAVA application, etc.

Please amend paragraph [0020] to read as follows:

[0020] Line 3 indicates the required prerequisite(s) in order to process the OML document. In example #1, IBM WebSphere® Portal Server 4.1.4 is required. This program will act as the web service to perform the parsing and node selection functions

required in order to generate the desired object.

Please amend paragraph [0022] to read as follows:

[0022] The OML definition is a very high level description of an application in comparison to the complete coding approach that was required in the prior art. The OML document is supplied to a web service that will parse the document and determine the best available model from all models residing within the grid or grids serviced by the web service for creating a complete application in accordance with the OML definition. In the example set forth, IBM WebSphere® Portal Server 4.1.4 is the selected web service.

Please amend paragraph [0025] to read as follows:

[0025] The style sheet parses the OML document using ~~Xerces~~ java XERCES JAVA parsing from ~~apache.org~~ APACHE.ORG (line 5), which is a well known parsing technique. The output file is not to be parsed (line 6). The output file is defined to be a Java JAVA file (line 7), and is to be encoded using ISO-8859-1 (line 8), which is a well known codepage.

Please amend paragraph [0027] to read as follows:

[0027] Once the OML description has been transformed into a fully coded application using the XSL stylesheet, it is output to an output file (in the example, to a java JAVA file). This file is then returned to the programmer. In the exemplary

PATENT

Application No. 10/733,037

Docket No. RSW920030159US1

Page 6

embodiment, the file is returned via the web service (e.g., WebSphere® Portal Server

4.1.4).